IT/97/11

STK system acceptance tests

Xavier Ferré Grau

December 4, 1997

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Objectives

This document describes the acceptance tests performed on the STK robotic system, which are part of the Tape Automation Project. This project is being pursued by the PDP (Physics Data Processing) group in the Information Technology division at CERN.

The CERN Computer Centre supports a distributed data storage service with primary storage on disk pools consisting of sets of Unix file systems, and secondary storage on a variety of magnetic tape subsystems. The basic model is that the data is permanently resident on magnetic tape and is accessed by the application through a disk cache. The data in the tapes is accessed through tape servers (Unix machines). Each tape server is connected to one or more tape drives and deals with transfers from drives to the network.

The user accesses data through the SHIFT software, which is a DST (Data Summary Tape) Analysis Facility developed at CERN. SHIFT was created, starting in 1990, to provide batch computing services on inexpensive RISC processors, with fast access to large amounts of disk data, and good tape support.

The service is changing, as the total requirement for storage of active data is estimated to double in the period 1996-1998. HPSS (High Performance Storage System), a new data management system, will be tested at CERN in the period 1997-1998. HPSS has been designed to provide a very big bandwith in data processing, and we expect that it will offer the flexibility and performance that next generation CERN experiments need.

Against this background, it was decided to acquire a fully automated magnetic tape storage system capable of handling all of the active data storage and access requirements foreseen until the year 2000. Fully automating the active data storage will make it possible to reduce substantially the manual operation coverage, which currently accounts for 50% of the cost of the service, and at the same time improve the average access time. A call for tender was issued in June 1996, calling for an automated tape system with the following characteristics:

- At least 300 tape fetch/mount/load/unload/dismount/return cycles per hour.
- At least 400 TBytes of total capacity.
- At least 20000 automated tape volumes (or scalable to that).
- At least 32 independent tape units. each capable of supporting a sustained data transfer rate of at least 8.5 MBytes/second (memory to tape).
- At least 50 MBytes/second as aggregate sustained data transfer rate.
- At least 10 GByte capacity for an individual cartridge.
- Remote diagnostic facilities.

STK (Storage Tek) submitted the best offer according to the adjudication criteria, so it was decided to purchase an STK automated tape storage system. The equipment consists of 3 Powderhorn silos with robotic arms to fetch the tapes, and 32 Redwood drives (views of a drive are shown in pages 22 and 23). Drives are connected through fast, wide differential SCSI to tape servers.

The present work describes the acceptance tests, which have been run on the STK system to verify that it accomplishes the requirements stated in the contract signed between CERN and STK. The features to test were:

- Cartridge Import/Export rate.
- Data Transfer rate.
- Cartridge fetch, mount and dismount rate.

The acceptance period has served not only to measure the actual performance of the STK equipment, but has also been a good test to evaluate the quality of the technical and support service provided by STK.

1.2 Contents

In the second chapter we detail STK system acceptance tests, with a description of the software developed for that purpose and results obtained.

In the third chapter we describe tests which are out of the scope of the acceptance procedure. They test maximum cartridge capacity and the influence of block size on data transfer rate.

1.3 STK system description

The STK automated tape library is composed of the following parts:

- 3 Powderhorn silos: Each one is an automated cartridge subsystem, and they are interconnected by a "pass-through" port. They are also equipped with an import-export feature to allow the exchange of cartridges with the outside. The configuration of February 1997 includes two silos, and a third one being installed now; in service on January 1998.

- Drive frames: 4 of them in the present configuration (February 1997). Each frame holds 4 CTUs (Controller Transport Units) and a SSU (Subsystem Support Unit).

- SSU: It supports and monitors the subsystem hardware. It is equipped with functional and diagnostic software, including a hard disk storing a small tape usage database. The SSU can be accessed through a serial port to download information from the database.

- CTUs - Redwood drives: Each CTU includes a drive. They are connected to tape servers through a fast wide diferential SCSI bus which can provide up to 11.1 MBytes/sec from data buffer to tape (uncompressed data).

The system configuration is shown in figure 1.1, and a view of the system in the Computer Room in building 513 is shown in figure 1.2.

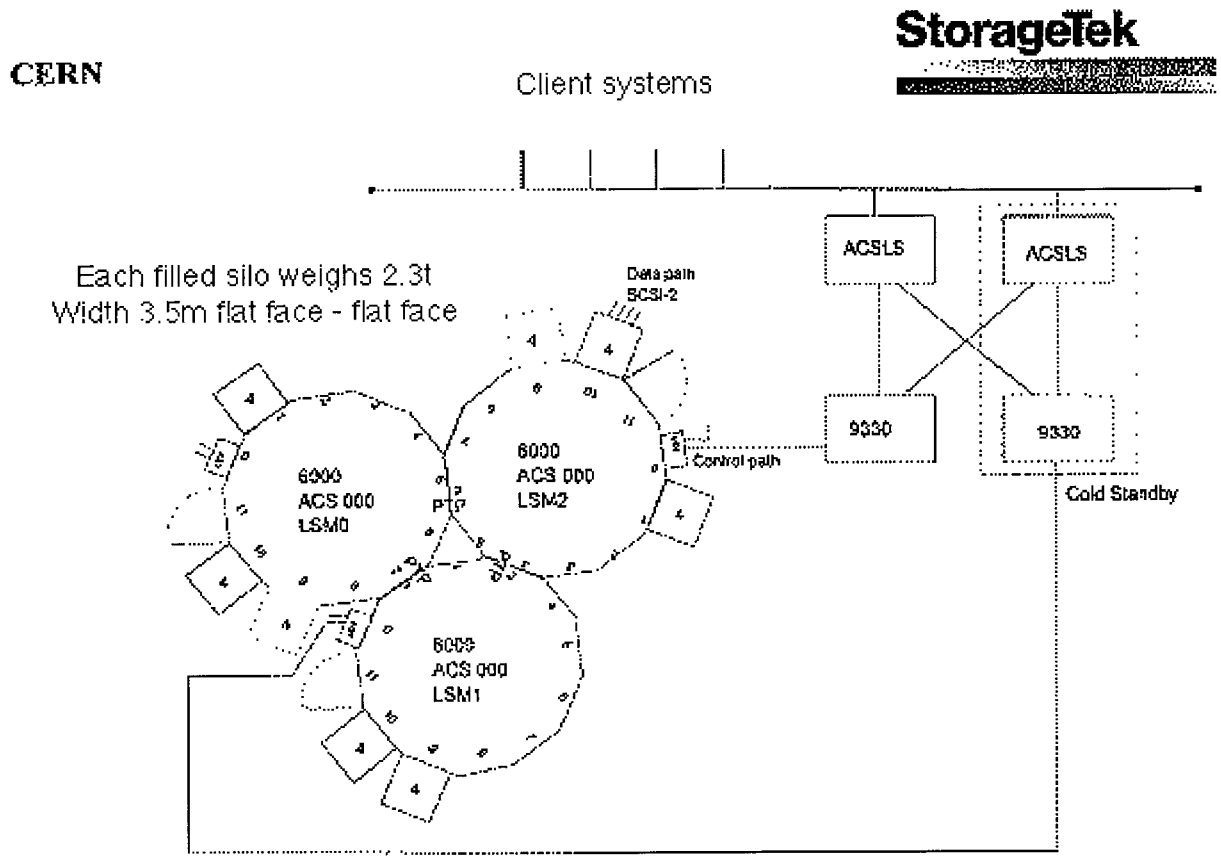

Figure 1.1: STK system configuration

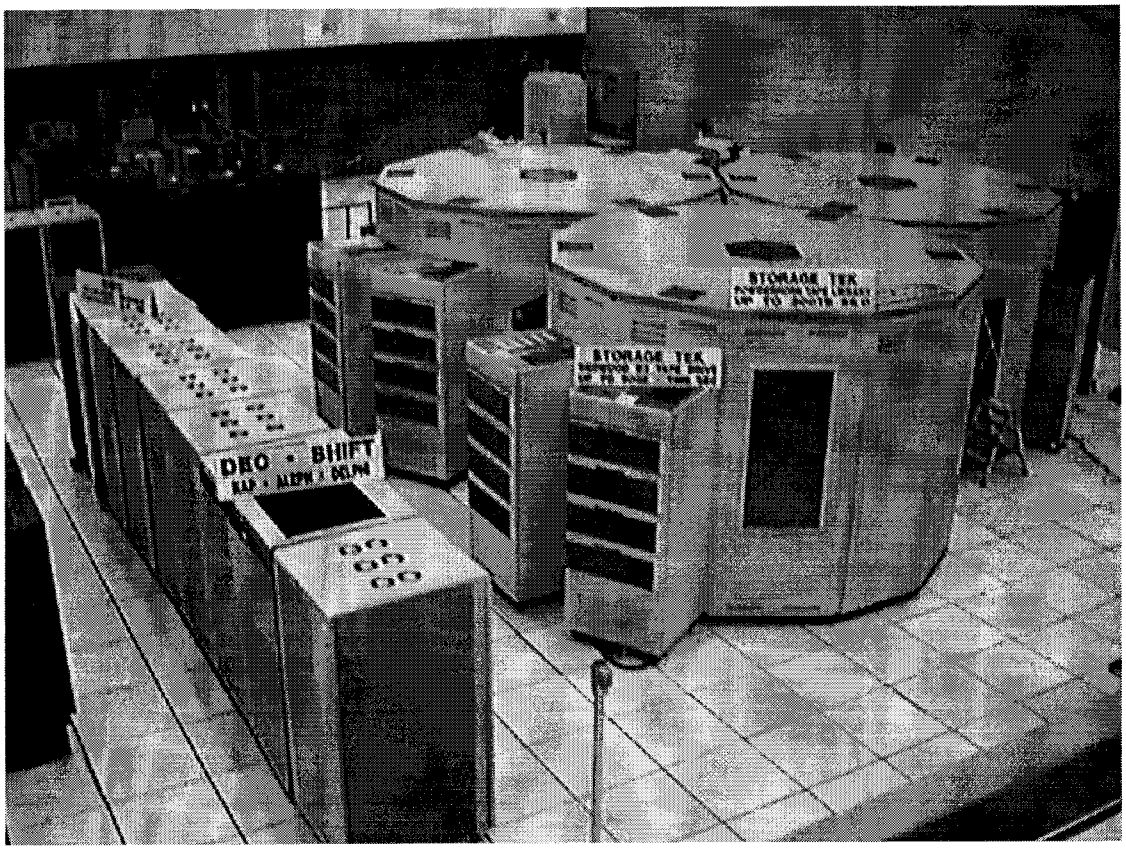

Figure 1.2: View of the STK robotic system and the tape servers attached to it

Chapter 2

STK system acceptance tests

2.1 Test descriptions

The tests measure three performance rates stated in the contract signed between STK and CERN:

- Data Transfer Rate: Sustained transfer rate between server memory and tape. This set of tests measures the reading/writing error rate as well.
- Cartridge Insert/Eject rate: How many cartridges can the robot import or export per hour from the automatic library.
- Maximum cycle rate: Testing both the robot and the drives, running as many streams as available (all 16 drives in the acceptance period configuration) to saturate drives by merely viewing the first 80 bytes of each tape (the VOL1 tape label). Each operation in the cycle includes the following steps:
  1. Picking the tape.
  2. Mounting it on a drive.
  3. Tape load (positioning to BOT[1]).
  4. Reading the first 80 bytes.
  5. Dismounting the tape (tape unload).

[1]Beginning Of Tape

6. Picking the tape to take it back to a library cell (not necessarily the original one).

All the tests have been carried out using Perl scripts. They are briefly described in the following sections, and their source code can be found in the WWW address *http://wwwinfo.cern.ch/~xavier/scripts.html.*

To store the results of any type of test a small database has been designed, in order to make the access to the data gathered simpler and to allow us treat the data in a structured way.

2.1.1 Test result database

Database design

The entity-relation model is shown in figure 2.1. This model can be simplified and translated to a smaller model. In this reduced model (figure 2.2) there are only two main entities: the volume and the operation. It is like that because a volume is a separate entity that we want to keep, as it has a physical meaning and having it makes easier data treatment. The rest of entities and relations in figure 2.1 can be packed in a single entity called Operation.

The database is indexed by volume and time stamp of the operation. We assume that there is no more than one operation requested over the same volume in the same second.

To implement it physically we use an SDBM database, as it can be easily managed from a Perl script and we don't require a good compression rate or fast data access.

The fields of each entry are the following ones:

- Keys:

  VolumeID Cartridge on which the operation has been performed.

  Date Time stamp corresponding to the beginning of the operation.

- Other fields:

  Operation_type It can be one of the following:

  m Mount.

  d Dumptape.

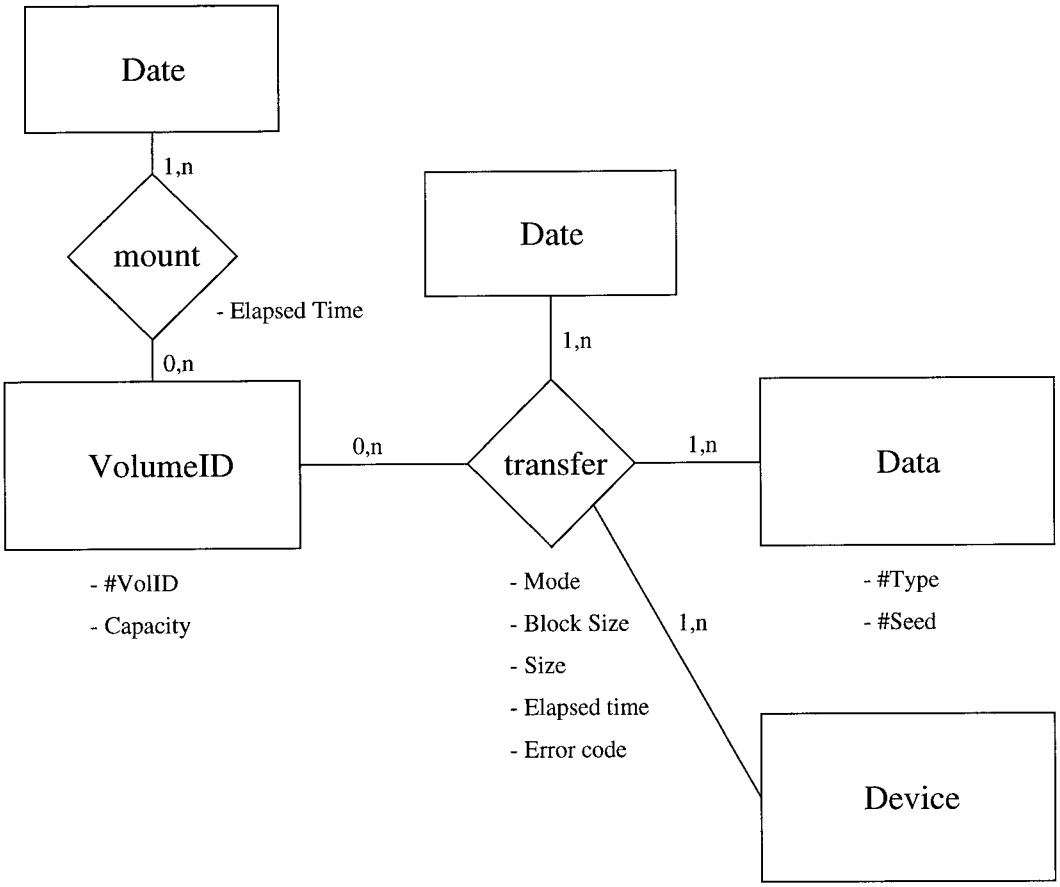

Figure 2.1: Entity-Relation model for test result database

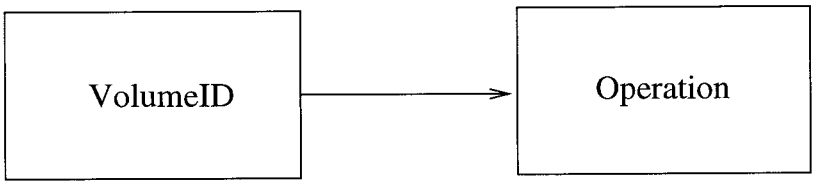

Figure 2.2: Simplified model for test result database

r Read.

w Write.

Elapsed_time Time spent in the operation.

Error_code If the operation has been completed successfully its value is zero. Otherwise, the different error codes are shown in tables 2.2 and 2.3.

Device The drive in which the operation has been performed. This field is empty for dumptape operations, and those where a device wasn't specifically stated.

Block_size Block size in bytes used in a transfer operation.

Size Size in MBytes (1 MByte = $2^{10} * 2^{10}$ bytes) of the data read or written in a transfer operation.

Data_type Type of data transferred. It can be one of the following (ordered by expected increasing compression rate):

C Random characters.

F Floating point numbers (sequential).

I Integers (sequential).

1 A sequence of blocks containing 20 zeros and 1 one.

0 All zeros.

Seed When data type is random characters, this value is the seed used to generate them. It makes it possible to repeat the same sequence for checking purposes.

Each database is stored in a pair of files in the form db_name.dir and db_name.pag

To access the data in these database files, the script `read_results.pl` has been written. Through this program we can calculate the average time for mountings, its typical deviation, the average total data transfer rate, the average transfer rate for each block size used, etc.

A call to this program can be like the following one:

```
read_results.pl -db db3.pag db2.pag -sort date
```

Its output is in an eight-column format, as shown in figure 2.3. The headings of each column correspond to the fields of the database as shown in table 2.1.

```
 Volid Date                      T  Time Er BSz GByte D Seed Device
 ----- ----                      -  ---- -- --- ----- - ---- ------
Y20601 Sat Aug  2 13:46:57 1997 w  5123  0  32 43.25 C   30 sd3r0090@shd50
Y20602 Sat Aug  2 15:26:39 1997 w  5149  0  32 43.50 C   30 sd3r0090@shd50
Y20603 Sat Aug  2 16:56:01 1997 w  5125  0  32 43.25 C   30 sd3r0090@shd50
Y20604 Sat Aug  2 18:24:54 1997 w  5154  0  32 43.50 C   30 sd3r0090@shd50
9.41744419873698 MBytes/sec average over 15 good data transfers.
Typical deviation: 0.627206805368067
          0 MBytes read    (    0.00 GBytes,  0.00 TBytes).
     669696 MBytes written (  654.00 GBytes,  0.64 TBytes).

BSz  Avge Tr Rate Typ Dev Number
---  ------------ ------- ------
 32  8.6519174313 0.01033      6
 64  9.8338369366 0.00270      3
256  9.9747745971 0.00454      6
```

Figure 2.3: Output from read_results.pl

Heading	Database field
Volid	VolumeID
Date	Date
T	Operation_type
Time	Elapsed_time
Er	Error_code
BSz	Block_size / 1024
GByte	Size / 1024
D	Data_type
Seed	Seed
Device	Device

Table 2.1: Correspondence between read_results.pl output headings and database fields

2.1.2 Error codes

The error codes of an operation performed in a test are specified in tables 2.2 and 2.3.

When an error code has a zero in its left part, it means that it has been supplied by the script `run_transfers.pl`, because it could not get the error code from the program that actually performed the transfer (the result was not returned properly in the auxiliary result file).

Error Code	Meaning
0	No error
1	Dumptape failed (it returned something different to zero)
2	The tape does not have an ASCII label (it is probably not labelled)

Table 2.2: Error codes for dumptape tests

2.1.3 Test launcher scripts

To make test execution easier and repeatable some scripts have been developed to control the execution of the real test scripts.

They are all based on the fact that tape servers each have several drives attached. Thus more than one test process can be run in parallel on each tape server. When there are as many test processes as drives, 100% usage of drives is achieved.

Depending on the way drive reservation is done, there are two launcher scripts:

group_test.pl Reserves the drives for all the jobs it is going to launch at a time, and when all its children end it releases the drives kept.

rsh_test.pl It launches each job through a remote shell (`rsh`), so all of them have a different Parent Process Id. Children are responsible for drive reservation and release. It doesn't wait for its children to finish.

Both programs read from a configuration file the parameters to launch a specific set of tests.

Error Code	Meaning
0	No error
1	Error reading or writing the tape (either sysread or syswrite failed)
2	The block read from the tape differs from expected
3	Drive releasing (command `rls`) took more than 10 minutes, so it is probably stuck
6	The error code does not appear in the expected format in the auxiliary result file
7	Cannot open auxiliary result file
8	The program received a SIGINT or SIGQUIT signal
9	A `die` Perl instruction was executed
10	`reserv` command failed (non zero returning)
11	`tpmnt` command failed (non zero returning)
12	Read command (`dd`) failed (non zero returning)
13	A release command keeping the reservation (`rls -k`) failed (non zero returning)
14	A normal release command (`rls -a`) failed (non zero returning)
15	The tape is still loaded when `transf_db.pl` returns
16	Error writing Trailer Label on tape
17	Auxiliary result file does not contain the header line we wrote, so it is probably corrupted

Table 2.3: Error codes for mount and transfer rate tests

```
# Script to run
script /u/c3/xavier/perl/mounts_db.pl

# Working dir
dir /u/c3/xavier/mount/int_52

# Number of processes concurrently executing the script
processes 4

# Parameters for the scripts which are common to all of them.
common_pars -rep 2

# Parameters for process 0
specific_pars 0 -dev sd3r0190 -o 0 -log log_0 -db db0 -l STK_0.lst
# Parameters for process 1
specific_pars 1 -dev sd3r0191 -o 1 -log log_1 -db db1 -l STK_1.lst
# Parameters for process 2
specific_pars 2 -dev sd3r0192 -o 2 -log log_2 -db db2 -l STK_2.lst
# Parameters for process 3
specific_pars 3 -dev sd3r0193 -o 3 -log log_3 -db db3 -l STK_3.lst
```

Figure 2.4: Example of configuration file for group_test.pl

```
# Script to run
script /u/c3/xavier/perl/run_transfers.pl

# Host
host shd49

# Working dir
dir /u/c3/xavier/trans/block_size/shd49

# Number of processes concurrently executing the script
processes 4

# Parameters for the scripts which are common to all of them.
common_pars -prog /u/c3/xavier/perl/transf_db.pl -no_skip_bad

# Parameters for process 0
specific_pars 0 -o 0 -par tr_par0 -log log_0 -db db0 -rep 8
# Parameters for process 1
specific_pars 1 -o 1 -par tr_par1 -log log_1 -db db1 -rep 8
# Parameters for process 2
specific_pars 2 -o 2 -par tr_par2 -log log_2 -db db2 -rep 8
# Parameters for process 3
specific_pars 3 -o 3 -par tr_par3 -log log_3 -db db3
```

Figure 2.5: Example of configuration file for rsh_test.pl

An example of configuration files for `group_test.pl` and `rsh_test.pl` are shown in figures 2.4 and 2.5.

Here we have an example of a call to group_test.pl:

```
group_test.pl -log group_log group_pars
```

And a call to rsh_test.pl:

```
rsh_test.pl -log rsh_log rtr_pars
```

To make the execution of this programs independent of the window they were eventually launched, they are run through a remote shell. The actual calling to the script is stored in a shell script called `run`, with a content like the following one:

```
#!/bin/sh
$HOME/perl/rsh_test.pl -log rsh_log rtr_pars
```

And the actual command entered is:

```
rsh shd51 "cd trans/int_51; run" < /dev/null &
```

We have then an execution independent of the parent session. So, even if the session where the execution began ends, the tests would go on running without any disturbance. This is specially useful when tests must run for days, as they can be started and the session can be closed without worrying about possible power cuts in the terminal or accidental log-off procedures.

2.2 Data transfer rate

The main objective of this set of tests is to measure the data transfer rate between tape server memory and a Redwood drive, as it is the first time that CERN works with this type of drive. But they also measure the hardware writing/reading reliability (number of writing/reading errors not recovered or not reported) and the influence of high drive activity on the transfer rate of drives attached to the same server.

To do these tests, two scripts have been developed:

run_transfers.pl Runs transfer tests with the specific parameters read from a file. The transfer is not performed directly by this program: it calls another script to do every transfer, and it reads the result from a file to store it in a test database. An example of a configuration file is shown in figure 2.6, and a typical call to it could be:

```
run_transfers.pl -prog /u/c3/xavier/perl/transf_db.pl
  -no_skip_bad -o 1 -par tr_par1 -log log_1 -db db1 -rep 8
```

transf_db.pl Performs a transfer (either read or write). It measures the time spent in the operation, counting only the time of the transfer, and not the time of the mounting/positioning/releasing. This script is usually called from `run_transfers.pl`. It provides a high flexibility, as it accepts a wide range of parameters to establish data size, block size, data type, etc.

When a read operation is done, it is checked that data matches what would be written with such parameters.

To be able to write labels when writing to a tape, a Perl interface for some SHIFT library functions has been developed. In order to run `transf_db.pl` it is necessary to include in the command line the location of this Perl extension library. So, a call to this program will be similar to the following one:

```
/usr/local/bin/perl5 -w -I/u/c3/xavier/perl/SHIFT_tapes/blib/arch
 -I/u/c3/xavier/perl/SHIFT_tapes/blib/lib
 /u/c3/xavier/perl/transf_db.pl -w -v Y00012 -g 9 -b 32 -d C
 -log /tmp/xavier/run_transfers.pl.23287 -o 0
 -out /tmp/xavier/aux_result.0 -s 30 -l -dev sd3r0090
 -reserv -tpm -d 10GC
```

2.2.1 Test results

Tests were performed directly on a tape server, keeping busy all the four drives attached to it with the same operation parameters (data size, data type, block size, etc). Data was transferred between tape server memory and the tape, without passing through disk.

```
# Testing the influence of block size on speed.
# For that purpose 50 GByte cartridges will be used to have
# the higher possible amount of time the machine
# transferring information to/from the drive.
# The cartridges used will be: Y10606 and Y10607.
# Always random characters will be written, to have more
# precise measuring times (as random characters are almost
# uncompressible).

# 1.- Writing data in 32 KByte and 64 KByte block sizes
w Y10606 22 32 C -s 30 -l -dev sd3r00A2 -reserv -tpm -d 50G
w Y10607 22.5 64 C -s 30 -l -dev sd3r00A2 -reserv -tpm -d 50G
# 2.- Reading data written in 1.
r Y10606 22 32 C -s 30 -l -dev sd3r00A2 -reserv -tpm -d 50G
r Y10607 22.5 64 C -s 30 -l -dev sd3r00A2 -reserv -tpm -d 50G
```

Figure 2.6: Example of configuration file for run_transfers.pl

They were carried out from the 25th of March to the 14th of April, in different intervals on the drives attached to tape servers shd50, shd51 and shd52.

Random characters were written to achieve the lowest possible compression ratio (to get more accuracy in the measurements), having five different sets of random characters to avoid that last-transfer-buffering could affect the results. The maximum block size was used: 256 KBytes. To minimise the time spent in mounting and dismounting tapes (in spite of it is not included in the elapsed time) 43 GBytes were read or written in each operation.

The total number of data transfers performed is 1152, half of them reads and the other half writes. Thus the total amount of data transferred is 47.25 TBytes, with 23.625 TBytes written and the same quantity read.

Not one of the readings failed, so all the data read matched what we wrote. It gives us an error rate of under one error per 23.65 TBytes written. The contract stated a maximum of one error per 10^{12} bits ($\approx$ 116 GBytes) written[2], which is largely verified by this test.

The average data transfer rate achieved is 10.5 MBytes/s with a typical

[2]Hard errors at a rate of no more than 1 in 10^{12} bits when reading back data previously recorded on the equipment which did not then notify any error

deviation of 0.48, which satisfies the requirement of at least 8.5 MBytes/s.

These tests were done using the four drives of each tape server at a time. Another test was carried out on a tape server, using only one of the drives (sd3r01A0@shd51) while the other three were idle. We could then measure the possible impact of high activity on same server drives in data transfer rate.

5 GBytes of data were transferred in 256 KByte blocks in each operation. A total of 8 writings and 8 readings were performed, and the average transfer rate achieved was 10.59 MBytes/s with a typical deviation of 0.023. As a consequence, drives don't appreciably drop in their performance when the other drives on the same server are in full use. It is probably due to the relatively high performance of the tape servers, which are DEC Alphaserver 4100 machines with 4 processors, 256 MB memory and a fast, wide differential SCSI connection to each drive.

2.3 Cartridge Insert/Eject rate

Robotic equipment performance is tested in this section of the tests.

The exchange of cartridges with the automated library is done through the CAP (Cartridge Access Port). It is shown in figure 2.7.

To carry out these tests we have used two sets of 21 cartridges. This is the maximum number which can be introduced at a time into the CAP. We have inserted and ejected the two sets alternatively, measuring the time the robot spends doing each operation (insertion or ejection) and the time we spend placing/taking out the cartridges from the CAP.

The manual insertion/extraction of cartridges was done as fast as possible, to reduce the influence of human manipulation on the test result. It should be noticed that when operators have to insert/extract tapes in a production scheme, they will probably not be as fast.

The procedures followed in each case are the following:

- Insertion

  To insert cartridges into a silo it is only necessary that the "CAP ENTER" indicator is on, then CAP door must be opened and the cartridges placed in the slots. When the door is closed the gripper reads the bar codes of all the cartridges and places them one by one in silo cells. When it finishes CAP is unlocked and the "CAP ENTER" indicator

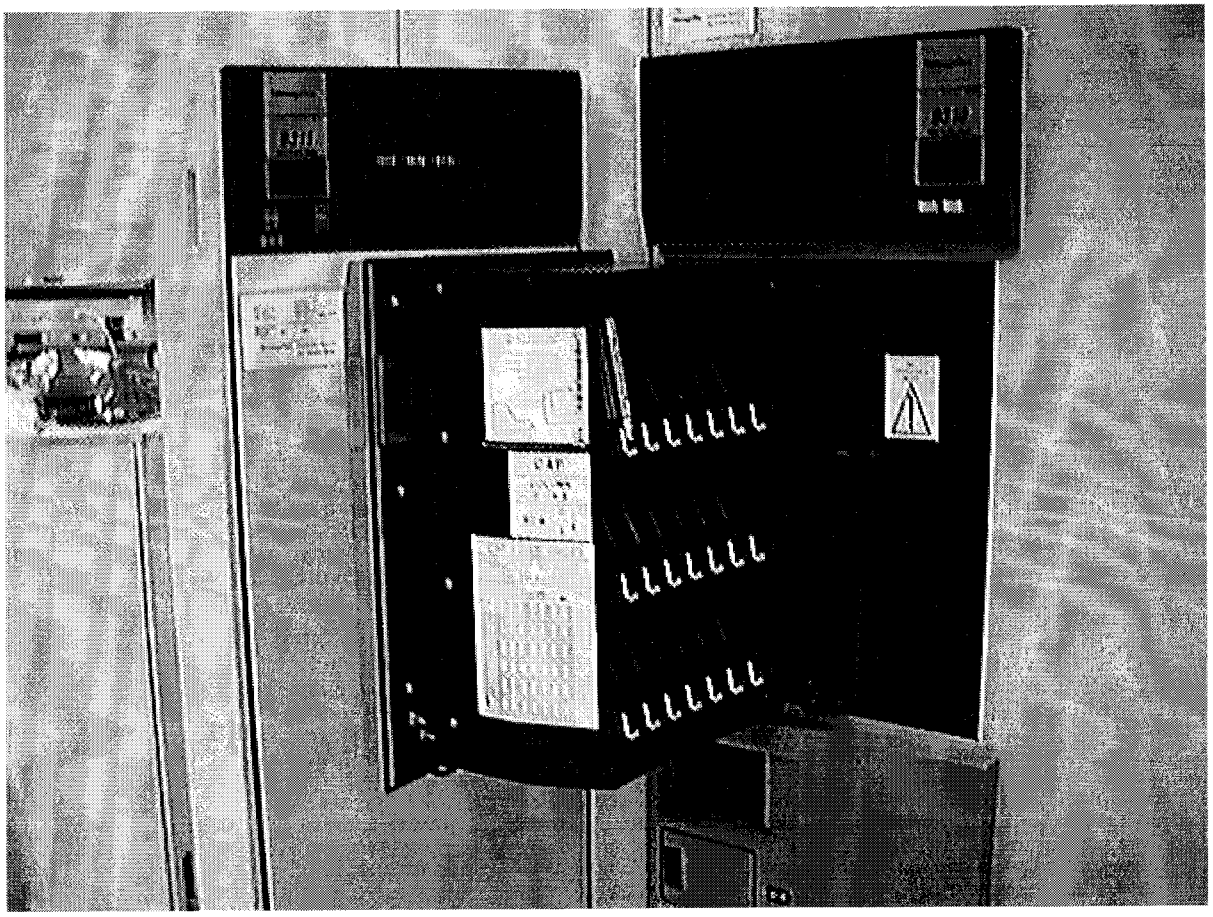

Figure 2.7: Cartridge Access Port (CAP)

is lighted on again. The time spent in all these operations is shown in the column named **Importing** on table 2.4.

- Ejection

  To eject a set of cartridges a command must be issued into the ACSSA facility. For example, to eject the set of volumes from Y20601 to Y20621, we should type:

  ```
  eject 0,1,0 Y20601-Y20621
  ```

  Then, the robot places, one by one, the requested cartridges into silo 1 CAP door. When it finishes the "CAP EJECT" is lighted on. It means that the request has been completed and the cartridges should be removed from the CAP door.

  Once the CAP has been emptied and it has been closed again, the robot checks the door contents and then lights on the "CAP ENTER" indicator.

  In the tests we have measured two steps in each ejection:

  - **Exporting**: From the moment when the eject command is issued to the "CAP EJECT" indication.
  - **Taking out cartridges**: CAP door emptying.

2.3.1 Test results

Both sets of cartridges were imported and exported 5 times from/to the robot. The average times for each operation are shown in table 2.4.

Importing	Exporting	Taking out cartridges
267.6	208.8	72.7

Table 2.4: Average time in seconds for cartridge Insertion/Ejection

We can get from this data which is the number of cartridges that the system can insert and eject per hour. The results are:

- Insert rate: 282 cartridges/hour.
- Eject rate: 268 cartridges/hour.

This is clearly over the minimum stated in the contract: 40 cartridges/hour.

2.4 Maximum cycle rate

Both the robotic equipment and the Redwood drives are tested in this set of tests.

Speed and reliability of the robotic equipment is tested, as it is pushed to a very high sustained cartridge request rate, which provokes an intensive use of the mounting/dismounting features and the "pass-through" mechanism.

On the other side, the amount of time taken by Redwood drives to thread and position to BOT (Beginning Of Tape) is also included in the "cycle" rate that is measured with these tests. Making an intensive use of each drive also provides us with an initial estimate about its reliability.

As it is described in page 6, we understand as "a cycle" the process of picking the tape, mounting it into the drive, positioning to BOT, reading several bytes, dismounting it and returning it back to a cell.

For this purpose tests consist of a loop of mounting commands over different volumes, which are read from a volumes list in a file. The mounting can be managed directly (with low level commands as `tpmnt`) or done through a `dumptape` command, so we have written two different scripts:

dumps_db.pl It uses `dumptape` with parameters `-N 1 -F 1`, to get only the first block of the first file. The inconvenience of using this command is that it always reads the first file of the tape completely (no matter what its size may be).

mounts_db.pl It mounts the tape using `tpmnt`, reads only the first 80 bytes using `dd` and releases the drive, to start again with the next volume.

2.4.1 Drive reliability test

`dumps_db.pl` is used to test the reliability of drives, since we don't care about the time spent in each operation. The characteristic to test is the number of mounts that a drive can do between mount/dismount failures. We used all the available tapes in the STK library, except a few ones which were in use for other tests. A total of 1518 cartridges were used for this test.

The objective is to do over 3000 mounts per drive. Thus each tape server runs four `dumps_db.pl` jobs, whose lists of volumes contain around 375 entries and each list is disjoint with the others (so they don't interfere). This test cannot give us great accuracy in the figures, because the drives had been used already before the test (so the number of mounts before a failure if any

is not exactly known) and with `dumptape` you cannot select the specific drive on which the mount will be carried out. However, running 4 simultaneous jobs in the same server can provide us with approximate figures.

Views of a Redwood drive are shown in figures 2.8 and 2.9.

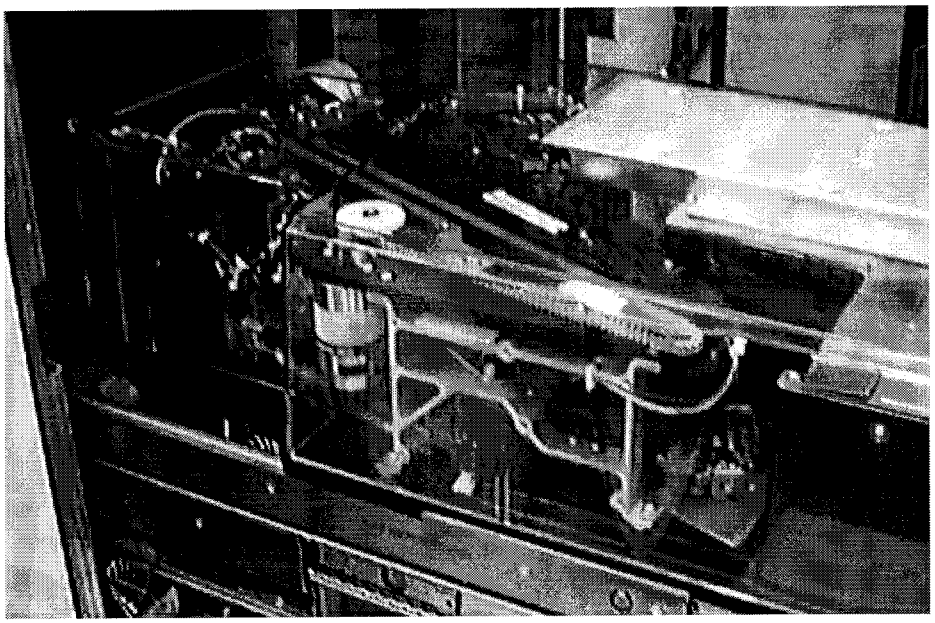

Figure 2.8: Redwood drive: Lateral view

Test results

Tests were carried out between the 5th and the 20th of March.

On tape server shd50 (frame 0,0,9) they run smoothly, doing all the mounts (more than 3000 per drive) without any drive problem.

On shd49 (frame 0,0,10) there was a read problem in the mount number 2703 with cartridge Y20001. As it was a particular cartridge problem, it cannot be counted as drive problem. As the other 3 jobs in the server did more than 3000 mounts, we can say that the test was successful in this server.

On shd51 there was an error on drive sd3r01A0. It was a dismount problem reported as `LIBRARY_FAILURE` with cartridge Y20886. It happened after this job had done 1023 mounts.

On shd52 there was one job which had an error on its mount number 1586, but it was `VOLUME_NOT_IN_LIBRARY`. This cartridge (Y10176) had been

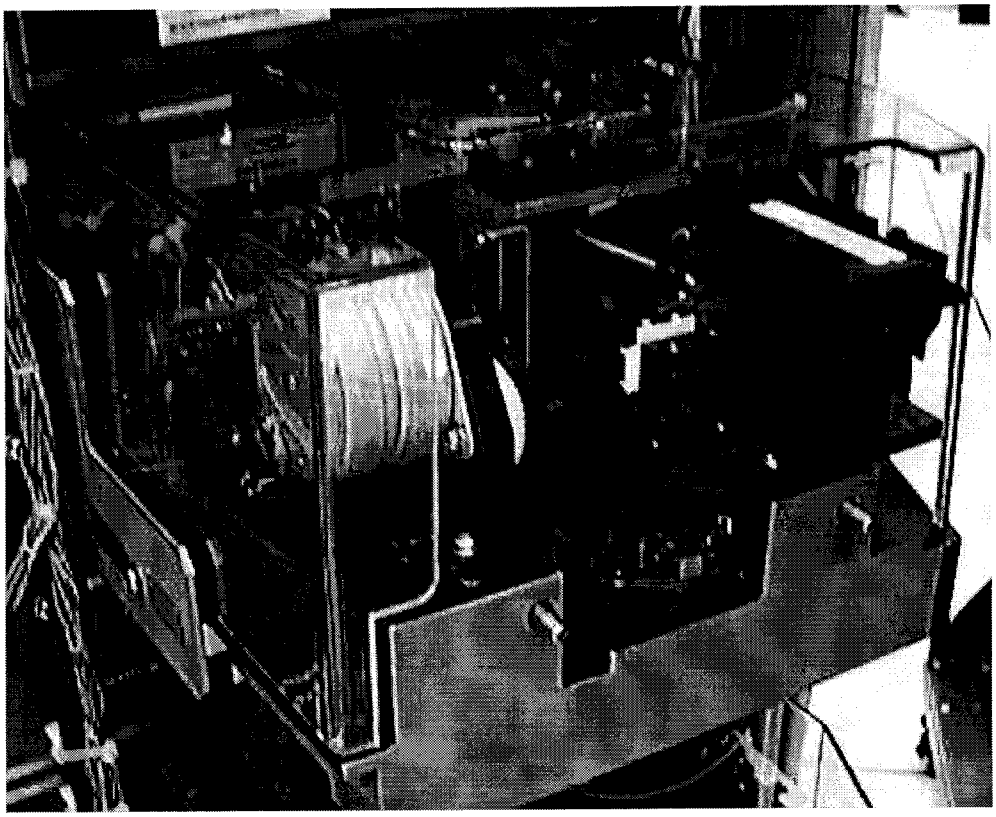

Figure 2.9: Redwood drive: Frontal view

exported from the silo due to a problem with another drive (with a job external to these set of tests), so it was not at all a problem of the drive we were testing. The job was restarted and did the rest of the mounts correctly. The other three jobs in this server ran smoothly doing their 3000 mounts.

We had only one drive problem in the execution of 41602 `dumptape` commands on 16 drives.

This gives us a rate of 1 drive problem every 2566 mounts. This is a good figure, better than the ones given by other robotic tape equipment currently running at CERN.

2.4.2 Cycle rate test

To test the maximum cycle rate we use the `mounts_db.pl` script, running one per available drive, all of them simultaneously. So we have 16 mounting processes issuing requests to the robot continuously for an hour.

344 cycles were performed in an hour. So, even with the acceptance test configuration (two silos instead of three), the figure in the contract is accomplished: minimum cycle rate of 300.

We can expect that with the final configuration we will achieve a higher cycle rate, but current results cannot easily be scaled for the final configuration due to the difficulty in evaluating the influence of cartridge exchanges between the three silos ("pass-through" facility).

We consider that this is an upper limit for the performance we can ask of the robotic equipment, as any job in a production scheme will read/write more than 80 bytes. Thus cartridges will stay longer in drives, leaving the robotics not as busy as in the test.

The average time taken by a cycle is 169 seconds, with a typical deviation of 46. The high typical deviation is due to the incidence of "pass-through" port exchanges between both silos, and due to peak moments when more than one mounting/dismounting request was issued at a time.

2.5 Problems found creating the tests

Running long-lasting tests directly on tape servers has been a difficult task. You must use the plain facilities provided by Unix system to run jobs, because the batch system at CERN is designed to run on a cernsp server and not

directly on a tape server. If you are going to use the low level tape commands you cannot use the batch system.

Tape drives are shared resources and the access to them is managed through the `reserv` command. Before mounting any tape the job must reserve all the drives it will use at the same time, so the system can tell if the job will be able to complete or not (in reference to the number of drives it will use). Drives must be released (through the `rls` command) when the job has finished with the mounted tapes. Reservations are assigned to process groups instead of being assigned to individual processes.

This scheme has caused us some problems with remote background jobs and drive reservation.

2.5.1 Running remote background jobs

For long-term running jobs (several days) it is not possible to have them executing on a X-terminal, as the cernsp (or whatever machine) server you are running on can eventually crash and all your jobs would go away. Therefore, you cannot have your jobs running in the foreground. You must execute them in the background to keep them running in case the window manager host dies.

But a normal background execution is not good either. Tape software classifies the jobs according to the Parent Process Id, and if the shell you run the script from dies, the job is inherited by the root process and it gets a Parent Process Id of zero. If you have several jobs in different windows they will all get Parent Process Ids of zero if their window die. This generates undesired effects, because they then belong to the same process group and the reservation of drives can be messy.

To avoid all these problems, test scripts have been launched through a remote shell. Every job has a different parent (the remote shell process), and the calling shell can be killed safely.

2.5.2 Drive reservation and others

There is an option to release a drive keeping the reservation because the job is going to mount another tape. We found that when multiple drives were reserved and we tried to release each drive at a different time, you could not release keeping the reservation because only one drive succeeds and the other ones stuck on the `rls` command. This has been solved in the

current release of SHIFT software, but the problem was initially avoided in our scripts through the usage of remote shell, so every job was independent of the others and did its own reservations (which is not the intended usage of the `reserv` command).

In addition, you must be very careful when running simultaneous jobs: They should not be running in the same directory, because they can eventually use temporary files with the same name. To solve this, our scripts run on a dedicated directory under `/tmp/xavier`.

Finally, this execution scheme made it difficult to stop a test in a clean way (not just killing it), so the scripts look before each mount if a file called `stop_test2` exists on its running directory (for a job executing on drive 2), and in that case the test stops smoothly.

2.6 Overall conclusions

The figures obtained for the 3 main points are the following:

- Data transfer rate (memory to tape): 10.5 MBytes/s
  - Less than 1 writing/reading error per 23.65 TBytes written.
- Insert rate: 282 cartridges/hour.

  Eject rate: 268 cartridges/hour.
- Maximum cycle rate: 344 cycles/hour.

The acceptance tests have been successfully completed by the system, so CERN has formally accepted it.

The acceptance period has served to push the system to an upper limit of its possible usage at CERN, as the number of mounts has been higher than it will be in production. It has provoked several drive unload failures, so we have been able to evaluate the technical service provided by STK. However, drive failure rate is inside reasonable limits. The robotic part of the system has a high reliability and we have been impressed by its general performance.

The three main points tested have offered us satisfactory results, despite the fact that the complete system with three silos is not yet available, so the principal objective of the tests has been achieved.

Chapter 3

Other tests

This chapter presents a description of tests performed on automated tape equipment, which do not belong to the formal acceptance tests for the STK system.

3.1 Cartridge capacity

When we began with transfer rate tests, we noticed that we could not write the amount of data nominally specified as the cartridge capacity. We obtained a *No space left on device* write error.

This led us to perform a set of tests to find out the real capacity of the Redwood cartridges we are working with. A 50 Giga cartridge is shown in figure 3.1.

In these tests we tried to write the nominal data capacity in each type of cartridge (10, 25 and 50 GBytes), writing random characters to avoid any automatic compression. As we obtained the "No space left on device" error, we lowered the amount of data written. Once we could write our tapes safely (without reaching the end of tape), we finished the test.

In figure 3.1 we can see the minimum amount of data with which we can get a "No space left on device" error, so it is an upper limit to the real cartridge capacity.

We contacted STK about this problem and they replied that the nominal capacity of a cartridge is counted measuring a KByte as 1000 bytes. So, for them, 1 GByte is 10^9 bytes.

Figure 3.1: 50 GB Redwood cartridge

Cartridge Nominal Capacity	Block Size	Minimum "unsafe" Capacity	Capacity Loss	% of Capacity Loss
10 GBytes	32 KBytes	9 GBytes	1 GByte	>10%
	64 KBytes	9 GBytes	1 GByte	>10%
	128 KBytes	9 GBytes	1 GByte	>10%
	256 KBytes	9.25 GBytes	0.75 GBytes	>7.5%
25 GBytes	32 KBytes	22 GBytes	3 GBytes	>12%
	64 KBytes	22 GBytes	3 GBytes	>12%
	128 KBytes	22 GBytes	3 GBytes	>12%
	256 KBytes	22.5 GBytes	2.5 GBytes	>10%
50 GBytes	32 KBytes	43.75 GBytes	6.25 GBytes	>12.5%
	64 KBytes	43.75 GBytes	6.25 GBytes	>12.5%
	128 KBytes	43.75 GBytes	6.25 GBytes	>12.5%
	256 KBytes	44. 5 GBytes	5.5 GBytes	>11%

Table 3.1: Cartridge capacity test results (1 GByte = $2^{10} * 2^{10} * 2^{10}$ bytes)

Even taking 1 GByte as 10^9 bytes we found that the real cartridge capacity is smaller than the nominal, as shown in table 3.2.

This limitation in cartridge capacity can be a problem when cartridges are meant to be used very close to the end of tape, specially if the data is already compressed when written to the cartridge. It can be a problem as well when copying directly 10 GByte DLT2000 cartridges into 10 GByte Redwoods, as some data might not fit onto the destination cartridge.

3.2 Influence of block size on data transfer rate

While doing transfer rate test, it was also observed that the block size has a big influence on transfer rate. With this set of tests we try to measure this influence to be able to provide users with information.

To carry out these tests we use the scripts `rsh_test.pl`, `run_transfers.pl` and `transf_dp.pl`, described in chapter 2.

50 GByte cartridges have been used, to maximise the amount of time the

Cartridge Nominal Capacity	Block Size (1 KByte = 1024 bytes)	Minimum "unsafe" Capacity	Capacity Loss	% of Capacity Loss
10 GBytes	32 KBytes	9.67 GBytes	0.33 GBytes	>3.36%
	64 KBytes	9.67 GBytes	0.33 GBytes	>3.36%
	128 KBytes	9.67 GBytes	0.33 GBytes	>3.36%
	256 KBytes	9.94 GBytes	0.06 GBytes	>0.67%
25 GBytes	32 KBytes	23.63 GBytes	1.37 GBytes	>5.51%
	64 KBytes	23.63 GBytes	1.37 GBytes	>5.51%
	128 KBytes	23.63 GBytes	1.37 GBytes	>5.51%
	256 KBytes	24.16 GBytes	0.84 GBytes	>3.36%
50 GBytes	32 KBytes	46.98 GBytes	3.02 GBytes	>6.04%
	64 KBytes	46.98 GBytes	3.02 GBytes	>6.04%
	128 KBytes	46.98 GBytes	3.02 GBytes	>6.04%
	256 KBytes	47.79 GBytes	2.21 GBytes	>4.43%

Table 3.2: Cartridge capacity test results (1 GByte = 10^9 bytes)

drive is transferring data. Uncompressed mode is used to avoid influence of compression rate, and to avoid any other possible compression we transfer random characters.

In table 3.3 and figure 3.2 are displayed the results of these tests.

It is clear that a block size of 64K or higher should be recommended to users in order to get a better performance.

32K , currently the most used block size, gives a good transfer rate but it can be improved with higher block sizes.

Block Size	Average Transfer Rate	Typical Deviation	Number of Tests
0.5K	0.348 MBytes/s	0.01297	34
1K	0.708 MBytes/s	0.00650	27
2K	1.395 MBytes/s	0.05306	31
4K	2.514 MBytes/s	0.07158	34
8K	4.364 MBytes/s	0.17272	31
16K	6.797 MBytes/s	0.25171	48
32K	9.445 MBytes/s	0.40450	29
64K	10.593 MBytes/s	0.00642	41
128K	10.6 MBytes/s	0.00551	18
256K	10.6 MBytes/s	0.00573	18

Table 3.3: Influence of block size on data transfer rate

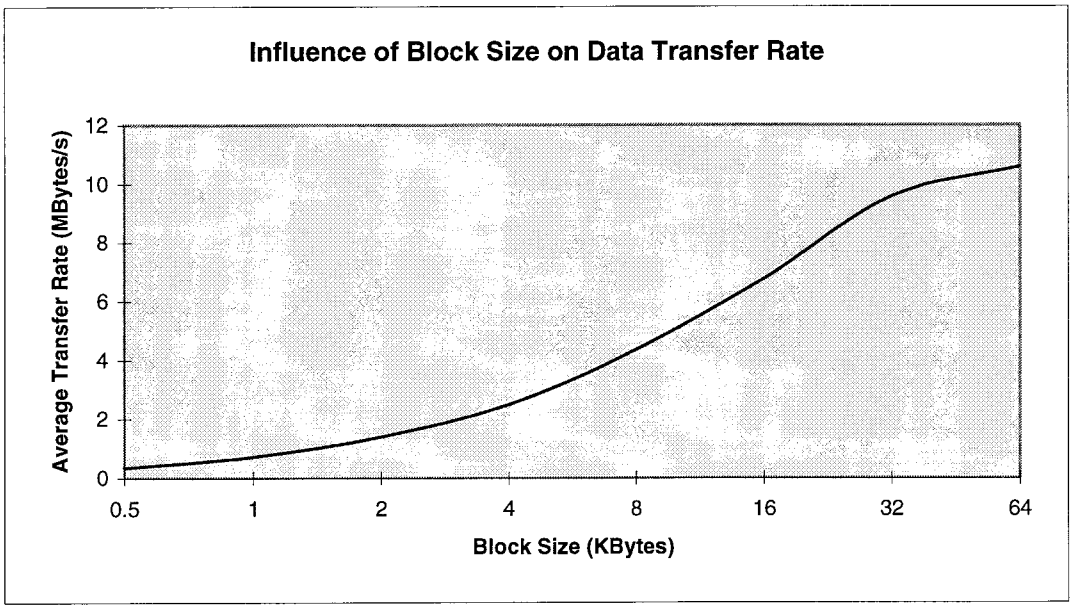

Figure 3.2: Influence of block size on data transfer rate